



IEEE Toronto Section

serving members and the community since 1903



Better Firmware... *Faster!*

A One-Day Seminar

Fri, April 20th, 2007

In Mississauga

Microchip Technology Regional Training Centre

6285 Northam Drive, Suite 100
Mississauga, ON L4V 1X5

Presented by **Jack Ganssle**, technical editor of *Embedded Systems Programming Magazine*, author of *The Art of Developing Embedded Systems*, *The Art of Programming Embedded Systems*, *The Firmware Handbook*, and *The Embedded Systems Dictionary*

Step1. Registration:
www.toronto.ieee.ca

IEEE Members: \$99
Non-Members: \$199

Step2. Payment:

Send cheque or money order to:

IEEE Toronto Section
2855 Bloor St. West
Suite 615, Toronto
ON, M8X 3A1

Contact:
d.cecic@ieee.org

This seminar will teach you new ways to build higher quality products in half the time.

80% of all embedded systems are delivered late...

Sure, you can put in more hours. Be a hero. But *working harder is not a sustainable way to meet schedules*. We'll show you how to plug productivity leaks. How to manage creeping featurism. And ways to balance the conflicting forces of schedules, quality and functionality.

... yet it's not hard to double development productivity

Firmware is the most expensive thing in the universe, yet we do little to control its costs. Most teams deliver late, take the heat for missing the deadline, and start the next project having learned nothing from the last. Strangely, *experience* is not correlated with *fast*. But *knowledge* is, and we'll give you the information you need to build code more efficiently, gleaned from hundreds of embedded projects around the world.

Bugs are the #1 cause of late projects...

New code generally has *50 to 100 bugs* per thousand lines. Traditional debugging is the *slowest* way to find bugs. We'll teach you better techniques proven to be up to 20 times more efficient. And show simple tools that find the nightmarish real-time problems unique to embedded systems.

... followed by poor scheduling

Though capricious schedules assigned without regard for the workload are common, even developers who make an honest effort usually fail. We'll show you how to decompose a product into schedulable units, and how to use killer techniques like Wideband Delphi to create more accurate estimates.

Learn From The Industry's Guru

Spend a day with Jack Ganssle, well-known author of the most popular books on embedded systems, technical editor and columnist for *Embedded Systems Programming*, and designer of over 100 embedded products. You'll learn new ways to produce projects *fast* without sacrificing quality. This seminar is the only non-vendor training event that shows you *practical* solutions that you can implement *immediately*. We'll cover technical issues – like how to write embedded drivers and isolate performance problems – as well as practical process ideas, including how to manage your people and projects.

Seminar Leader



Jack Ganssle has written over 300 articles in Embedded Systems Programming, EDN, and other magazines. His three books, *The Art of Programming Embedded Systems*, *The Art of Developing Embedded Systems*, and his most recent, *The Embedded Systems Dictionary* are the industry's standard reference works

Jack lectures internationally at conferences and to businesses, and was this year's keynote speaker at the Embedded Systems Conference. He founded three companies, including one of the largest embedded tool providers. His extensive product development experience forged his unique approach to building better firmware faster.

Jack has helped over 600 companies and thousands of developers improve their firmware and consistently deliver better products on-time and on-budget.

Course Outline

Languages

- C, C++ or Java?
- Code reuse – a myth? How can you benefit?
- Stacks and heaps – deadly resources you *can* control.

Structuring Embedded Systems

- *Manage* features... or miss the schedule!
- Do commercial RTOSes make sense?
- Five design schemes for faster development.

Overcoming Deadline Madness

- Negotiate realistic deadlines... or deliver late.
- Scheduling – the science versus the art.
- Overcoming the biggest productivity busters.

Stamp Out Bugs!

- Unhappy truths of ICEs, BDMs, and debuggers.
- *Managing* bugs to get good code fast.
- *Quick* code inspections that keep the schedule on-track.
- Cool ways to find hardware/software glitches.

Managing Real-Time Code

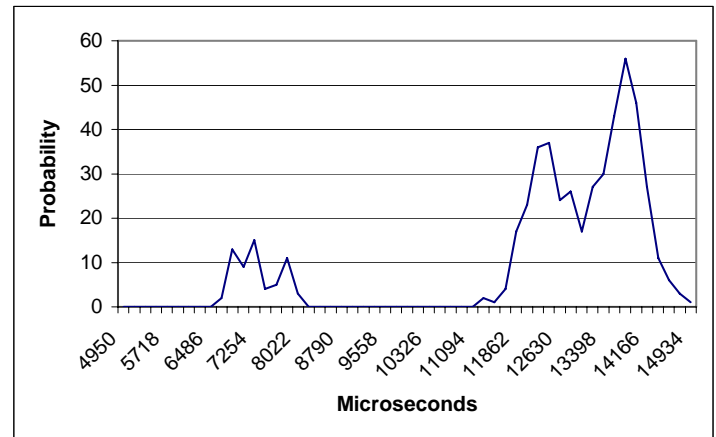
- Design *predictable* real-time code.
- Preventing system performance debacles.
- Troubleshooting and eliminating *erratic crashes*.
- Build better interrupt handlers.

Interfacing to Hardware

- Understanding high-speed signal problems.
- Building peripheral drivers faster.
- Cheap – and expensive – ways to probe SMT parts.

How to Learn from Failures... and Successes

- Embedded disasters, and *what we can learn*.
- Using postmortems to accelerate the product delivery.
- Seven step plan to firmware success.



Do those C/C++ runtime routines execute in a usec or a week? This trig function is all over the map, from 6 to 15 msec. You'll learn to rewrite real-time code proactively, anticipation timing issues before debugging.

Why Take This Course?

Frustrated with schedule slippages? Bugs driving you batty? Product quality sub-par? **Can you afford not to take this class?**

We'll teach you how to get your products to market faster with fewer defects. Our recommendations are *practical, useful today, and tightly focused* on embedded system development. Don't expect to hear another clever but ultimately discarded software methodology. You'll also take home a 150-page handbook with algorithms, ideas and solutions to common embedded problems.